

УДК 681.3.06

ОБОСНОВАНИЕ ЛИНЕЙНО-УПОРЯДОЧЕННОГО ПРЕДСТАВЛЕНИЯ ГРАФОВЫХ МОДЕЛЕЙ ПРОГРАММ

С.В. Огородов

Институт «Кибернетический центр» ТПУ

E-mail: serg@ad.cctpu.edu.ru

Предложено линейно-упорядоченное представление графовых моделей программ в виде совокупности ветвей и ветвей-связей, в основу которого положена новая двухпозиционная FK-нумерация вершин ориентированных графов. Данное представление графовых моделей программ позволяет полностью выявить все структурные характеристики управляющих графов программ и на их основе реализовать эффективные алгоритмы оптимизации объектного кода программ при их трансляции с языков программирования высокого уровня.

Графовые модели программ (ГМП), которые по своей сути являются ориентированными графами (орграфами), отображают структуры программ для ЭВМ. В ГМП в качестве вершин выступают операторы, процедуры и модули программ, а дугами являются передачи управления между ними. В соответствии с этим, основу ГМП составляет управляющий орграф (граф) программы $H(V, W)$ [1]. Здесь V – множество исполняемых вершин ГМП, W – множество управляющих связей (дуг) между вершинами множества V .

Управляющий граф H характеризуется следующими свойствами:

- имеется одна вершина, в которую не входит ни одна управляющая дуга, назовем ее входом в ГМП (v_0);
- каждая вершина графа $H(v_1, v_2, \dots, v_n)$ имеет один вход, на который может поступать несколько передач управления (управляющих связей) с других вершин, и один или два выхода (для логических операторов);
- имеется множество дуг W_k , назовем их выходами из ГМП, которые не имеют в качестве своих приемников ни одной вершины из множества V .

Второе свойство управляющего графа H является основой для упрощения алгоритмов анализа его структуры. При этом, если в реальных программах есть операторы, которые имеют несколько управляющих выходов, например m , то их в ГМП можно представить в виде цепочки $(m-1)$ вершин, каждая из которых имеет по два управляющих выхода. Для логических вершин ГМП примем, что первый выход из вершины соответствует ответу «да» на условие, проверяемое в операторе, а второй – ответу «нет». В соответствии с этим, обозначим множества управляющих дуг для первого и второго выходов всех вершин графа H через $W_1(H)$ и $W_2(H)$.

Для анализа структуры орграфов используются два стандартных метода: обход вершин графа в глубину и обход графа в ширину. В результате выполнения первого из них строится дерево простых путей в графе, а также выявляются множества древесных, прямых и обратных дуг [2]. Дерево простых путей при этом не является однозначным, состав вершин, входящих в «древесные» пути зависит от

стратегии обхода графа [3]. При этом класс обратных дуг образует в орграфе множество циклов, выявление которых является одной из основных задач анализа структуры управляющих графов программ [4]. Второй метод позволяет построить дерево кратчайших путей в графе [5]. Информация, получаемая в результате выполнения данных обходов вершин управляющих графов программ, практически никак не может быть использована для решения задач оптимизации объектного кода программ при трансляции исходных программ с любого языка программирования, представленных в виде ГМП, в программы на языке ассемблера конкретной ЭВМ.

В связи с этим, предлагается реализовать представление управляющего графа H в линейно-упорядоченном виде, в результате построения которого выявляются все циклы в ГМП и определяются уровни их вложенности, выявляются все линейные и параллельные участки ГМП, с учетом которых задачи оптимизации объектного кода могут решаться гораздо эффективнее [6]. Приведем ряд определений, раскрывающих понятия, используемые при описании линейно-упорядоченного представления ГМП, а также докажем три леммы.

Определение 1. Точками ветвления управляющего графа H называются вершины, из которых выходит по две управляющих дуги, каждая из которых ведет в еще не пройденные вершины.

Определение 2. Ветвью управляющего графа H называется последовательность вершин и дуг, получаемая при обходе графа в глубину до тех пор, пока не выполнится одно из четырех условий, при которых дальнейшее продвижение по вершинам графа становится невозможным. Данные условия состоят в следующем:

- дуга (или обе дуги), выходящие из конечной вершины ветви, принадлежат множеству W_k выходных дуг из ГМП;
- последняя из дуг, по которым рассматривалась возможность продвижения вперед, оказалась дугой, замыкающей цикл;
- последняя из рассматриваемых дуг дает вхождение в порождающую ветвь после ответвления;
- последняя рассматриваемая дуга дает вхождение в ветвь, порожденную ранее, чем текущая ветвь.

Первая ветвь графа H всегда начинается с начальной вершины v_0 . Последующие ветви начинаются с еще не пройденных вершин, которые определяются через стек, хранящий обратные отображения данных вершин, а именно, точки ветвления управляющего графа H . Условия, введенные в Определении 2, это условия окончания ветвей графа H .

Определение 3. Порожденными ветвями называются ветви, начинающиеся с вершин, имеющих своими обратными отображениями точки ветвления управляющего графа H . Порождающими ветвями называются ветви, которые имеют хотя бы одну точку ветвления.

Определение 4. FK -нумерацией вершин орграфа H назовем такую двухпозиционную идентификацию вершин орграфа, при которой каждой из вершин поставлена в соответствие пара $\{f_i, k_i\}$, где f_i — номер ветви, которой принадлежит вершина v_i , а k_i — есть ранг вершины v_i , равный длине простого пути из начальной вершины v_0 в вершину v_i , полученного при обходе графа H в глубину.

Определение 5. Циклом управляющего графа H называется последовательность вершин $v_i, v_{i+1}, v_{i+2}, \dots, v_{i+n}$, в которой вершина v_{i+n} имеет наибольший ранг и инцидентна дуге, ведущей в v_i . Дугу (v_{i+n}, v_i) назовем дугой, образующей цикл.

Определение 6. Ветвью-связью управляющего графа H называется дуга, соединяющая вершины, принадлежащие различным ветвям, или же — вершины одной ветви, между которыми существует путь, состоящий из вершин и дуг, или же дуга, образующая цикл.

Введем понятия явных и неявных ветвей-связей.

Определение 7. Ветвь-связь называется неявной, если она соединяет вершины порождающей и порожденной ветвей, имеющие ранги, отличные друг от друга на единицу.

Пусть $k_n(v_i)$, $k_o(v_j)$ — ранги начальной и конечной вершин v_i ветви-связи, v_i принадлежит порождающей ветви. Тогда условие $k_o(v_j) - k_n(v_i) = 1$ будет соответствовать обозначенному выше определению.

Определение 8. Ветвь-связь называется явной, когда $k_o(v_j) - k_n(v_i) \neq 1$, здесь v_i и v_j принадлежат либо одной ветви, если дуга (v_i, v_j) образует цикл или дает передачу управления вперед, либо различным ветвям вне зависимости от их порождаемости, а также тогда, когда начальная вершина v_i ветви-связи принадлежит порожденной ветви, ее конечная вершина v_j — порождающей ветви и $k_o(v_j) - k_n(v_i) = 1$.

Обозначим множества ветвей, явных и неявных ветвей-связей графа H через $\Phi(H)$, $\Theta(H)$ и $M(H)$ соответственно. Граф H в этом случае можно представить в виде объединения соответствующих множеств. Обозначим множество порождающих ветвей через $\Phi'(H)$, множество порожденных ветвей — через $\Phi''(H)$. Между $\Phi'(H)$ и $\Phi''(H)$ существует следующее соотношение: $\Phi'(H) \cap \Phi''(H) \neq \emptyset$. Это означает, что порождающая ветвь графа H может быть порождена другой ветвью этого графа, одна и та же

ветвь может принадлежать как множеству $\Phi'(H)$, так и $\Phi''(H)$. Лишь относительно первой ветви графа H можно сказать определенно, что $\varphi_1 \in \Phi'(H)$.

Количество ветвей в множествах $\Phi'(H)$ и $\Phi''(H)$ определяется в общем случае степенью сложности управляющего графа. Только в случае, когда множество дуг вторых выходов из вершин графа H является пустым ($W_2(H) = \emptyset$) или $W_2(H) \subset W_k$, то можно сказать, что $|\Phi'(H)| = 1$ и $|\Phi''(H)| = 0$. Эти равенства справедливы также тогда, когда выполняется условие: $\forall w_2(v_i) \in W_2(H) \exists (w_1(v_i) \in W_k)$, $i = 0, 1, 2, \dots, N-1$, где N — количество исполняемых вершин графа H , $w_1(v_i)$ и $w_2(v_i)$ — дуги, соответствующие первому и второму управляющим выходам из вершины v_i . Из Определения 7 неявных ветвей-связей следует, что их количество в управляющем графе H всегда равно количеству порожденных ветвей в данном графе. Значительно больший интерес представляет вопрос о количестве ветвей и явных ветвей-связей в графе H .

Лемма 1. Количество ветвей и ветвей-связей в управляющем графе H равно $|W_2(H)| - |W_2(H) \cap W_1(H)| \in W_k + 1$.

Доказательство. Предположим, что $W_2(H) = \emptyset$. Тогда $|W_2(H)| = |W_2(H)| - |W_2(H) \cap W_1(H)| \in W_k = 0$. Количество ветвей в этом случае будет равно единице. Этой единственной ветвью является ветвь, начинающаяся с начальной вершины v_0 . Если $|W_2(H)| > 0$, $|W_2(H) \cap W_1(H)| \in W_k > 0$ и $|W_2(H)| = |W_2(H) \cap W_1(H)| \in W_k$, то в этом случае количество ветвей также равно единице и эта ветвь начинается с v_0 . Если же $|W_2(H)| > 0$, $|W_2(H) \cap W_1(H)| \in W_k > 0$ и $|W_2(H)| > |W_2(H) \cap W_1(H)| \in W_k$, то в графе H при его обходе в глубину выявляются точки ветвления и/или передачи управления на уже пройденные вершины. При этом первоначальная ветвь, которой соответствует единица в доказываемом выражении, также есть ветвь, начинающаяся с v_0 . Каждая точка ветвления, по определению которой w_1 и w_2 ведут в еще не пройденные вершины, образуют одну ветвь или явную ветвь-связь. Каждая вершина, не являющаяся точкой ветвления и имеющая управляющие дуги w_1 и w_2 , одна из которых ведет в еще не пройденную вершину и продолжает текущую ветвь, а другая — передает управление в уже пройденную вершину, образует в соответствии с Определением 8, одну явную ветвь-связь. Следовательно, сумма вершин, имеющих w_2 , определяет общее количество ветвей и ветвей-связей. Вершины, у которых существуют w_2 и $w_1 \in W_k$, способствуют уменьшению общего количества ветвей и явных ветвей-связей, т. к. в этом случае вершина, являющаяся концом управляющей дуги w_2 , становится текущей вершиной формируемой ветви и w_2 не образует новую ветвь или ветвь-связь. Таким образом, $|W_2(H)| - |W_2(H) \cap W_1(H)| \in W_k + 1$ есть общее количество ветвей и явных ветвей-связей в управляющем графе H .

Определение 9. Линейно-упорядоченным представлением управляющего графа H называется его представление в виде совокупности ветвей-связей и ветвей максимальной длины.

Линейно-упорядоченное представление управляющего графа H характеризуется тем, что ранг k_i в FK-нумерации каждой вершины v_i графа H соответствует длине максимального простого пути из начальной вершины v_0 в вершину v_i . При этом, передача управления из некоторой вершины v_i в вершину v_j , для которых выполняется соотношение $k_j < k_i$, всегда является передачей управления на цикл.

Лемма 2. Для управляющего графа H , представленного в виде совокупности ветвей и явных ветвей-связей, полученных в результате обхода графа H в глубину, приведение его к линейно-упорядоченному виду не меняет суммарного количества ветвей и явных ветвей-связей.

Доказательство. В соответствии с леммой 1 количество ветвей и явных ветвей-связей в управляющем графе H определяется мощностью $|W_2(H)|$. Так как приведение графа H к линейно-упорядоченному виду не связано с изменением его управляющей структуры и, соответственно, не изменяет $|W_2(H)|$, то суммарное количество ветвей и явных ветвей-связей в графе H при таком преобразовании не изменяется.

Для построения линейно-упорядоченного представления управляющего графа H необходимо прежде всего выполнить обход вершин данного графа в глубину. Использование стандартного обхода, предназначенного для построения дерева простых путей в орграфах общего вида [2], является нерациональным для управляющего графа $H(V, W)$, вершины которого имеют полустепень исхода $\deg^-(v) \leq 2$. Поэтому автором разработан более эффективный алгоритм построения дерева простых путей графа H с одновременным выявлением циклических управляющих дуг в нем [3]. Эффективность предложенного алгоритма прохождения вершин орграфа, основанного на методе поиска в глубину, определяется следующим:

- при обходе графа запоминаются не дуги, а вершины;
- вершины проходятся только один раз в прямом направлении (кроме случая выявления циклических дуг, когда вершины, анализируемые на принадлежность циклу, проходятся в прямом направлении дважды);
- возвращение назад в случае невозможности продвижения вперед происходит не по пройденным дугам, как при стандартном обходе [2], а сразу в последнюю встретившуюся точку ветвления, занесенную ранее в стек.

В результате выполнения данного алгоритма формируется следующий вектор-характеристика для каждой ветви: $E_j = \{F_j, D_j, A_j, K_j^*, K_j^o, T_j, R_j, N_j, f_j^1, f_j^2, \dots, f_j^N\}$. В этом векторе: F_j – порядковый номер ветви; D_j – длина ветви, выраженная в количестве вершин, принадлежащих данной ветви; A_j – адрес начала ветви в оперативной памяти ПЭВМ; K_j^* , K_j^o – ранги начальной и конечной вершин ветви; K_j^o – ранг вершины, в которую входит ветвь F_j ; T_j – тип ее окончания; R_j – количество разветвлений в ветви; N_j – количество

прародительных ветвей для данной ветви (если $N_j = 0$, то прародительных ветвей для данной ветви нет); $f_j^1, f_j^2, \dots, f_j^N$ – номера прародительных ветвей.

Для каждой ветви-связи формируется вектор-характеристика из шести позиций: $E_j = \{F_j, 0, v_j^*, K_j^*, K_j^o, T_j\}$. Здесь: F_j – порядковый номер ветви-связи; 0 – признак вектора-характеристики ветви-связи (обозначает «длину» ветви-связи: она равна нулю, так как не содержит вершин графа H); v_j^* – порядковый номер вершины, из которой выходит ветвь-связь; K_j^* – ранг данной вершины (начальный ранг ветви-связи); K_j^o – ранг вершины, в которую входит ветвь-связь (конечный ранг ветви-связи); T_j – тип окончания ветви-связи.

Значение T_j в векторах-характеристиках соответствует условиям окончания ветвей при обходе графа H в глубину, которые были введены в Определении 2. В соответствии с этим, возможны четыре условия окончания ветвей, некоторые из которых влияют на возможность удлинения ветвей графа, построенных после его обхода в глубину, для формирования его линейно-упорядоченного представления. На рис. 1 приведены фрагменты управляющего графа H , отображающие все случаи, при которых возможно удлинение ветвей, построенных при данном обходе графа H . Рис. 1, а, б, показывают возможности удлинения ветвей I и III, соответственно, в случае, когда ветвь или ветвь-связь входит в ветвь, порожденную ранее (четвертое условие окончания ветви). Ветвь-связь III и на рис. 1, в, и ветвь I на рис. 1, г, а также ветвь II на рис. 1, д, соответствуют третьему условию окончания ветвей. Ветвь-связь III с рис. 1, в, дает удлинение ветви I за счет части ветви II. Ветвь I с рис. 1, г, удлиняется за счет части ветви II, присоединяемой к окончанию ветви I. Цифры рядом с вершинами на рис. 1 обозначают ранги вершин до удлинения ветвей.

Лемма 3. Необходимым условием удлинения ветвей управляющего графа H является наличие в нем ветвей или ветвей-связей, у которых $K_j^o \geq K_j^*$ и $T_j = 3$, или $T_j = 4$.

Доказательство. Если $T_j = 1$, то конечная управляющая дуга ветви F_j , выходящая из вершины v_j^* , принадлежит множеству W_k . Так как прямое отображение данной вершины является пустым, то есть $\Pi(v_j^*) = \emptyset$, то удлинение ветви в этом случае невозможно. Если $T_j = 2$, то конечная управляющая дуга w_j^* является дугой, замыкающей цикл. Для такой дуги по определению цикла $K_j^* > K_j^o$, то есть строгое выполнение отношения $K_j^* \geq K_j^o$ в данном случае является необходимым условием существования цикла, а не условием удлинения.

Доказательство необходимости условия $K_j^* \geq K_j^o$ при $T_j = 3$, или $T_j = 4$ для удлинения ветвей проведем способом от противного. Пусть $T_j = 3$ или $T_j = 4$ и $K_j^* < K_j^o$. Докажем невозможность удлинения ветвей в этом случае. Так как конечная вершина ветви F_j имеет при этом ранг меньший, чем вершина v_j^o , принадлежащая ветви F_j^o , в которую произошло вхождение ветви F_j , то это значит, что ветвь F_j^o длиннее F_j , и присоединение к F_j части ветви F_j^o уменьшит ранги вершин, принадлежащих этой части ветви F_j^o , следовательно, такая перестройка не

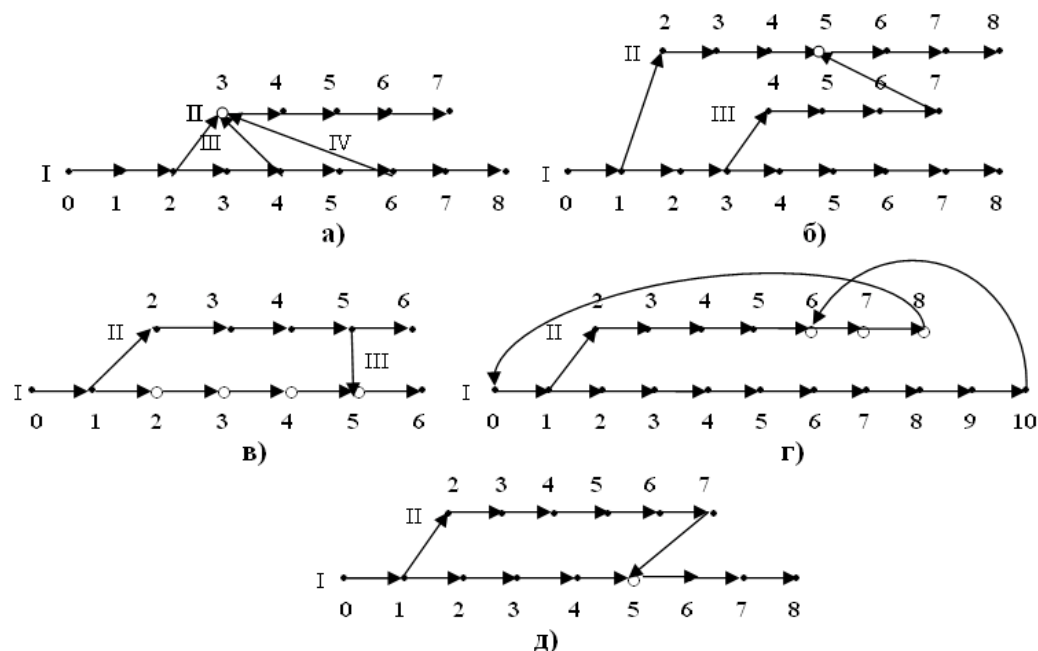


Рис. 1. Фрагменты управляющего графа H , отображающие возможные условия окончания ветвей в результате обхода графа H в глубину

будет удовлетворять определению линейно-упорядоченного представления графа H и $K_j^* < K_j^o$ не является условием, способствующим удлинению. Значит, необходимым условием для удлинения ветвей при $T_j=3$ или $T_j=4$ является противоположное условие $K_j^* \geq K_j^o$ и, тем самым, лемма 3 доказана.

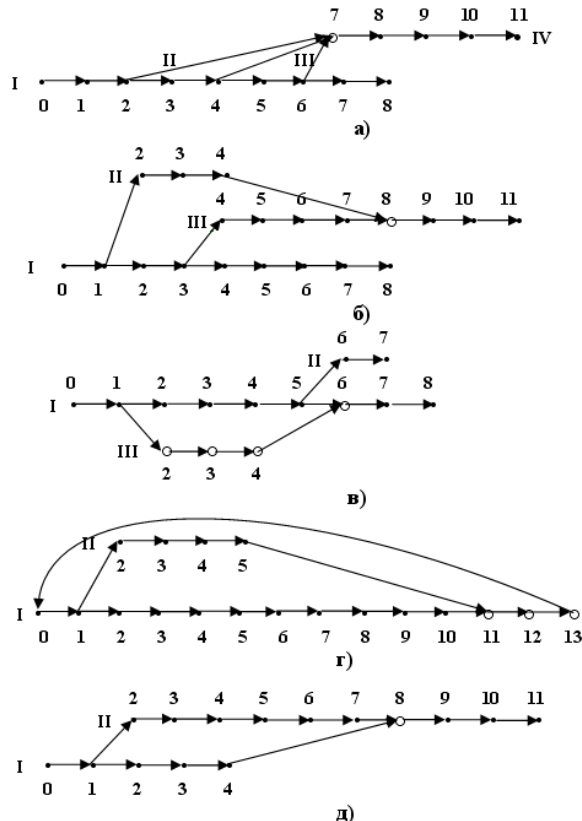


Рис. 2. Фрагменты управляющего графа H , представленные на рис. 1, после удлинения ветвей

На рис. 2 представлены те же фрагменты управляющего графа H , что и на рис. 1, но после соответствующего удлинения ветвей. Удлинение в случае рис. 1, а, назовем удлинением в начале ветви, так как здесь происходит «перенос» начала ветви вправо (начальной вершиной ветви IV становится вершина 0 с рангом $K_j^*=7$), которая до удлинения имела ранг 3 и принадлежала ветви II. Данное удлинение изображено на рис. 1, в, д. Удлинения в случаях, приведенных на рис. 1, б, г, назовем удлинением в середине ветви, так как здесь происходит удлинение одной ветви за счет другой ветви или ее части в середине удлиняемой ветви (рис. 2, в, д). Удлинения в случаях, показанных на рис. 1, б, г, назовем удлинением в конце ветви, так как здесь выполняется присоединение части некоторой ветви к окончанию удлиняемой ветви (рис. 2, б, г). Цифры рядом с вершинами на рис. 2 обозначают ранги вершин после удлинения ветвей. Кругами представлены вершины, по которым можно установить изоморфизм фрагментов графа H , представленных на рис. 1 и 2.

Представление управляющего графа H в виде совокупности ветвей максимальной длины и ветвей связей позволяет выявить в нем все циклы и определить уровни их вложенности, выявить все параллельные и линейные участки программ и создает тем самым все предпосылки для выполнения оптимизирующих преобразований на ГМП. При оптимизации циклов определяются вершины, которые могут быть вынесены из цикла вперед или назад, устанавливаются переменные, которые могут всего один раз загружаться из оперативной памяти на регистры до начала выполнения цикла. Параллельные и линейные участки программ учитываются при оптимизации распределения реги-

стров и распределения оперативной памяти ЭВМ, а также для выполнения некоторых других оптимизирующих действий при генерировании объектно-

го кода, в частности, для уменьшения исполнения команд безусловного перехода в объектных программах.

СПИСОК ЛИТЕРАТУРЫ

1. Погребной В.К. Об одном способе представления алгоритмов в виде графовых моделей // Управляющие системы и машины. – 1983. – № 1. – С. 63–69.
2. Евстигнеев В.А. Применение теории графов в программировании / Под ред. А.П. Ершова. – М.: Наука, 1985. – 352 с.
3. Огородов С.В., Трегубова Е.В. О структурном анализе графовых моделей алгоритмов // Методы и программы решения оптимизационных задач на графах и сетях. Ч. 1: Алгоритмы, программы, применения: Тез. докл. III Всесоюз. совещ. – Новосибирск: ВЦ СО АН СССР, 1982. – С. 156–158.
4. Касьянов В.Н. Анализ структур программ // Кибернетика. – 1980. – № 1. – С. 48–61.
5. Майника Э. Алгоритмы оптимизации на сетях и графах. – М.: Мир, 1981. – 323 с.
6. Огородов С.В. Задача автоматизации распределения регистров на графовых моделях алгоритмов // Методы и программы решения оптимизационных задач на графах и сетях. Ч. 1: Алгоритмы, программы, применения: Тез. докл. III Всесоюз. совещ. – Новосибирск: ВЦ СО АН СССР, 1984. – С. 169–171.

Поступила 14.04.2008 г.

Ключевые слова:

Ориентированные графы, графовые модели программ, управляющий граф программы, ветви и ветви-связей.

УДК 681.142.2

ЯЗЫК ОПИСАНИЯ ГЕНЕРАТОРОВ КОМБИНАТОРНЫХ МНОЖЕСТВ

А.В. Титков, В.В. Кручинин

Томский государственный университет систем управления и радиоэлектроники

E-mail: avt@tcde.ru

Предложен универсальный инструмент для построения и исследования генераторов комбинаторных множеств, основанный на функциональном языке, позволяющем описывать схемы рекурсивных композиций деревьев И/ИЛИ. Предложен вариант реализации такого инструмента, как расширение системной библиотеки STL для языка программирования C++.

Введение

Алгоритмы генерации комбинаторных множеств находят широкое применение при построении и исследовании разнообразных программно-технических систем. Например, в различных системах тестирования. Термин «программный генератор» носит достаточно общее толкование, например «генератор отчета», «генератор программного кода», «генератор случайных чисел» [1]. Здесь под генератором будет пониматься программа, реализующая некоторый алгоритм генерации комбинаторного множества. Комбинаторное множество представляет собой конечное множество, элементы которого имеют некоторую общую структуру. Например, множество перестановок, разбиений, двоичных деревьев, таблиц специального вида и т. д.

Различают 4 класса алгоритмов комбинаторной генерации [2]:

- 1) последовательная генерация множества всех комбинаторных объектов данного класса (*listing*);
- 2) нумерация всех комбинаторных объектов данного класса (*ranking*);
- 3) генерация комбинаторных объектов в соответствии с процессом нумерации (*unranking*);

- 4) случайная генерация реализации комбинаторного объекта (*random selection*).

Наиболее изучены алгоритмы последовательной генерации. Однако всё более востребованными являются генераторы, в основе которых лежат алгоритмы нумерации объектов и генерации объекта по его номеру.

Методов построения алгоритмов генерации и нумерации существует большое множество, и все они зависят от конкретного рассматриваемого комбинаторного объекта. Сравнительно недавно появился универсальный метод построения алгоритмов генерации и нумерации комбинаторных объектов, основанный на представлении комбинаторного множества в виде дерева И/ИЛИ [3].

В данной статье предлагается на основе метода построения и исследования алгоритмов генерации комбинаторных множеств универсальный программный инструмент построения и исследования генераторов элементов комбинаторных множеств.

Основные этапы и объекты метода

Метод построения алгоритмов генерации и нумерации комбинаторных объектов, основанный на